



USB in Depth

Session 116





USB in Depth

Craig Keithley
USB and FireWire Technology Evangelist

Introduction

- Discuss USB implementation issues facing developers
- Share tools and techniques that are helpful with creating USB drivers





USB in Depth

Rhoads Hollowell
USB Software Team

What You Will Learn

- USB documentation available
- Open Source and the IOUSBFamily
- IOService termination
- Code-less kernel extensions
- Dealing with kIOExclusiveAccess errors
- USB Prober for Mac OS X!
- Debugging techniques for USB drivers



Documentation

USB

- Start with Darwin documentation
<http://developer.apple.com/techpubs/macosx/Darwin/kernel.html>
- IO Kit fundamentals can be found at
<http://developer.apple.com/techpubs/macosx/Darwin/IOKit/IOKitFundamentals>
- Working With USB Device Interfaces
<http://developer.apple.com/techpubs/macosx/Darwin/IOKit/DeviceInterfaces/USBBook>
- USB Technology Home Page
<http://developer.apple.com/hardware/usb/index.html>



More Documentation

USB

- Technical Q&As
 - Tips on USB driver matching for Mac OS X
<http://developer.apple.com/qa/qa2001/qa1076.html>
 - Making sense of IO Kit error codes
<http://developer.apple.com/qa/qa2001/qa1075.html>
 - Issues with boot time KEXT loading
<http://developer.apple.com/qa/qa2001/qa1087.html>



Darwin and IOUSBFamily

- Darwin provides Open Source access to parts of the Mac OS X
- CVS module is IOUSBFamily (live!)
- Tags identify releases of module
 - IOUSBFamily-5Q125 used in build 5Q125 of Mac OS X 10.1.4
 - IOUSBFamily-6C35 used in current Jaguar build





Demo

Darwin Repository

**Fernando Urbina
USB Software Team**

Darwin and IOUSBFamily (Cont.)

- Use majordomo to get notices of checkins!
 - Send email to **majordomo@opensource.apple.com**
 - Body should include **subscribe cvs-log-iousbfamily**
- Use **usb@lists.apple.com** for questions
 - **http://lists.apple.com**
 - Search archive first
 - Always respond to entire list



IO Kit Convention for Apple Supplied Driver

- **IO**xxx (e.g., **IO**USBHIDDriver)
 - May be subclassed
- **Apple**xxx (e.g., **Apple**USBKeyboard)
 - Not intended to be subclassed
 - Not guaranteed to be binary compatible
 - Can borrow source for your own driver



IO Kit Termination Sequence

- Used when a device is unplugged
- Affects both Device and Interface drivers
- Method Declarations
 - willTerminate (IOService *provider, IOOptionBits options)**
 - didTerminate(IOService *provider, IOOptionsBits options, bool *defer)**
- No longer use kIOServiceMessageIsTerminated



IO Kit Termination Sequence (Cont.)

- **willTerminate()**
 - Driver **isInactive() == true**
 - Need to cancel or abort any outstanding I/O calls to provider



IO Kit Termination Sequence (Cont.)

- **didTerminate()**
 - Termination is almost complete
 - If all outstanding I/O is done, close the provider
 - If not, leave provider open and close it after the last I/O completes



Code-less Kernel Extensions

- You can have a kernel extension that only provides a personality
 - Add property to device nub
 - Provide vendor-specific driver so that class driver does not match to it
 - Use another driver as the binary
- Use Project Builder but have no code



Code-less Kernel Extensions

Example #1

- Want HID device available only to Classic
- Need a vendor-specific kext to match to device so IOUSBHIDDriver will not load for it
- Need to add “ClassicMustSeize” boolean property to device nub



Code-less Kernel Extensions

Example #1 (Cont.)

| | | |
|-----------------------------|------------|-------------------------------------|
| ▼IOKitPersonalities | Dictionary | ‡ 2 key/value pairs |
| ▼IOService driver | Dictionary | ‡ 5 key/value pairs |
| CFBundleIdentifier | String | ‡ com.apple.kernel.iokit |
| idProduct | Number | ‡ 281 |
| idVendor | Number | ‡ 1293 |
| IOClass | String | ‡ IOService |
| IOProviderClass | String | ‡ IOUSBDevice |
| ▼Merge driver | Dictionary | ‡ 6 key/value pairs |
| CFBundleIdentifier | String | ‡ com.apple.driver.AppleUSBMergeNub |
| idProduct | Number | ‡ 281 |
| idVendor | Number | ‡ 1293 |
| IOClass | String | ‡ AppleUSBMergeNub |
| IOProviderClass | String | ‡ IOUSBDevice |
| ▼IOProviderMergeProperties | Dictionary | ‡ 1 key/value pairs |
| ClassicMustSeize | Boolean | ‡ Yes |
| ▼OSBundleLibraries | Dictionary | ‡ 1 key/value pairs |
| com.apple.iokit.IOUSBFamily | String | ‡ 1.8.2 |
| OSBundleRequired | String | ‡ Root |



Code-less Kernel Extensions

Example #2

- Have a vendor-specific device
- Need to create interfaces for it
- Create a vendor-specific code-less kext that uses the AppleUSBComposite driver
 - Calls SetConfiguraton() which creates the interfaces
 - Handles reconfiguration after reset



Code-less Kernel Extensions Example #2 (Cont.)

| Property List | Class | Value |
|------------------------------------|---------------|---|
| CFBundleDevelopmentRegion | String | English |
| CFBundleExecutable | String | RainbowDemo |
| CFBundleIconFile | String | |
| CFBundleIdentifier | String | com.apple.iokit.RainbowDemo |
| CFBundleInfoDictionaryVersion | String | 6.0 |
| CFBundlePackageType | String | KEXT |
| CFBundleSignature | String | ???? |
| CFBundleVersion | String | 1.0.0 |
| ▼IOKitPersonalities | Dictionary | 1 key/value pairs |
| ▼Rainbow Dongle | Dictionary | 5 key/value pairs |
| CFBundleIdentifier | String | com.apple.driver.AppleUSBComposite |
| idProduct | Number | 768 |
| idVendor | Number | 1209 |
| IOClass | String | AppleUSBComposite |
| IOProviderClass | String | IOUSBDevice |
| ▼OSBundleLibraries | Dictionary | 1 key/value pairs |
| com.apple.driver.AppleUSBComposite | String | 1.8 |



Using USBLog() in Your KEXT

- Use instead of IOLog()
- Same printf style formatting
- Uses levels 1–7 to filter messages
- Sends a message to kernel logging KEXT
 - KLog.kext in SDK
 - Works if it is not there
- User space application gets that message and displays it (USB Prober)



Using USBLog() in Your KEXT (Cont.)

- Large buffer means no missed messages

```
# define DEBUG_LEVEL 3
```

```
# include <IOKit/usb/IOUSBLog.h>
```

- #define DEBUG_LEVEL (0-3) in your KEXT
- DEBUG_LEVEL of 0 causes USB Log to be stripped

- Sample usage

```
USBLog(3, "%s[%p]: USB Generic Composite @ %d",  
        getName(), this, _device->GetAddress() );
```





USB in Depth

Fernando Urbina
USB Software Team

Notifications of USB Plug and Unplug in User Space

- Need to know when devices come and go
- Need to distinguish between identical devices
- Look at DTS Sample at

http://developer.apple.com/samplecode/Sample_Code/Devices_and_Hardware/USB/USBPrivateDataSample.htm

- Shows how to include per device data in termination notifications



Dealing With Exclusive Access Errors

- User space applications need to open device or interface interfaces
- Used to arbitrate access to the USB Device or USB Interface
- Might get a `kIOReturnExclusiveAccess` error (0xe00002c5)
- Some other object has the device or interface open
 - Could be another KEXT
 - Could be another user client



Dealing With Exclusive Access Errors (Cont.)

- Use USB Prober to determine who has it open
- Look at IOService Plane

- A kernel extension

Apple Optical USB Mouse@2122000

AppleUSBComposite

IOUSBInterface@0

AppleUSBOpticalMouse

IOUSBUserClientInit

- A user client (probably Classic)

SNAPSCAN 1212U@2113000

IOUSBUserClientInit

IOUSBDeviceUserClient



Dealing With Exclusive Access Errors (Cont.)

- If a kernel extension (probably a class driver)
 - Vendor-specific code-less KEXT
 - Will match to device and return true from start method
- If Classic has it open
 - Code-less KEXT that has “ClassicMustNotSeize”
 - Better: Use **USBDeviceOpenSeize()** or **USBInterfaceOpenSeize()**



New Tools

- Jaguar KEXT tools
 - kextload does everything
- Logging KEXT (KLog.kext)
 - Found in USB SDK 1.8.7
 - Needed for USBLog() to work
- USB Prober
 - Just like old times, but better





Demo

USB Prober

**Nima Parivar
USB Software Team**

USB Prober

- Available now

**[ftp://ftp.apple.com/developer/Tool_Chest/
Testing_-_Debugging/Hardware_tool](ftp://ftp.apple.com/developer/Tool_Chest/Testing_-_Debugging/Hardware_tool)**

- Written in Cocoa
 - Prints
 - Cut and Paste
- Open-sourced (soon!)



KEXT Debugging Overview

- Two types of debugging
 - After the fact decoding of a panic
 - Active debugging (2-machine)
- In both cases, you need symbols
 - Build KEXT with symbols
 - Use kextload to generate symbol file
 - Locally or on another machine
 - Use addresses from panic message



Generating Symbols

- Build with symbols:

```
% pbxbuild install COPY_PHASE_STRIP=NO STRIP_INSTALLED_PRODUCT=NO  
INSTALLED_PRODUCT_ASIDES=YES STRIP=/usr/bin/true
```

- Results placed in build directory
and in install directory
- Use kextload to generate symbols

- If KEXT is already running

```
% sudo kextload -s /var/tmp -A Your.kext
```

- If KEXT is not running (asks for addresses)

```
% sudo kextload -n -s /var/tmp Your.kext
```



Panic Information

- New UI does not show panic addresses
 - Saved at `/Library/Logs/panic.log`
- If you want “old” style panic screen
 - **`sudo nvram boot-args="debug=0x104"`**
- Use addresses from module dependencies to generate symbols offline
- Use addresses from backtrace to find culprit!



Use gdb to Debug Your KEXT

- If machine is not available
 - Generate symbol file
 - **`gdb > add-symbol-file yourKext.sym`**
 - **`gdb > | *0x12345678`**
- If machine is available use 2-machine debugging
 - Look at “Hello Debugger (Debugging a device driver with GDB)” tutorial in developer site





Demo

KEXT Debugging

Resources

USB Implementers Forum

<http://www.usb.org>

Mac OS X Developer Information

<http://developer.apple.com/macosx/>

Macintosh USB Development Information

<http://developer.apple.com/hardware/usb/>



Roadmap

108 Managing Kernel Extensions:

Using IO Kit KEXTs in Mac OS X

Civic
Wed., 10:30am

515 Image Capture Framework:

Image Capture Framework

Room C
Fri., 2:00pm

808 Managing I/O: CFRunLoop and CFStream:

Using CFRunLoop in Applications

Room C
Wed., 2:00pm

FF009 FireWire and USB:

Tell us what you think

Room J1
Fri., 9:00am



Who to Contact

Worldwide Developer Relations

Craig Keithley

USB and FireWire Technology Evangelist

keithley@apple.com

USB Developer Mailing List

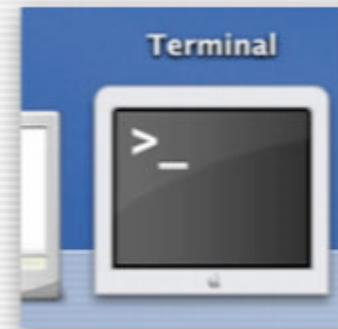
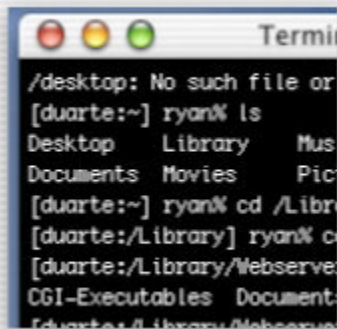
usb@lists.apple.com

<http://developer.apple.com/wwdc2002/urls.html>





Q&A



Craig Keithley
USB and FireWire Technology Evangelist
keithley@apple.com

<http://developer.apple.com/wwdc2002/urls.html>

 **WWDC2002**

 **WWDC2002**

 **WWDC2002**